BY: RAJA R.K.

Intrusion Detection with Snort Part–I
On the Alert

The July 2003 issue of LFY introduced many open source network security tools for system and network administrators. Snort was one of them. In this first article in a two-part series, we show you how to use Snort for intrusion detection.

Intrusion detection is the process of finding the attacks attempted against your computer or network of computers and responding to them in real-time. The software that performs this job is usually called an intrusion detection system (IDS). There are many free and commercial IDSs available in the market. One such tool is the Snort Intrusion Detection System.

Snort is an open source, light-weight network intrusion detection system. It has the ability to sniff, log and analyse network traffic in real-time.

Snort works on many platforms, including Linux, Windows and Solaris. It is licensed under GPL and available free of cost.

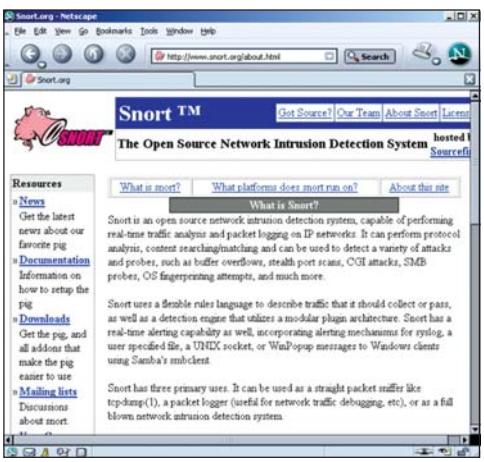


Figure 1: Snort—an open source network intrusion detection system

SNORT ARCHITECTURE

Snort has three major sub-systems: the packet decoder, the detection engine and the reporting sub-system. Snort uses libpcap library to sniff out packets from the network.

The packet decoder sub-system decodes each network packet in real-time. Snort currently supports Ethernet, SLIP and PPP protocols. The detection engine sub-system analyses the network packet against the set of Snort rules. Snort optimises the detection mechanism by grouping the rules into a two-dimensional structure of

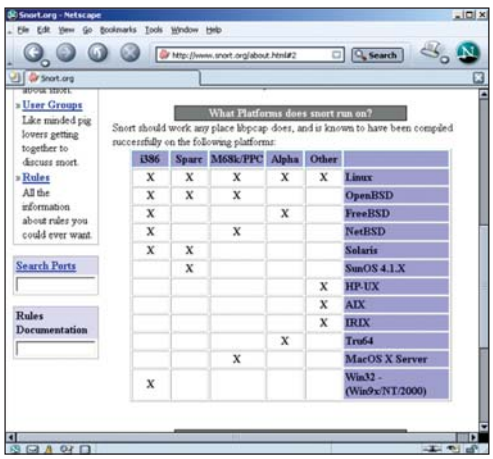


Figure 2: A truly cross-platform tool

Figure 3: Snort architecture (higher-level view)

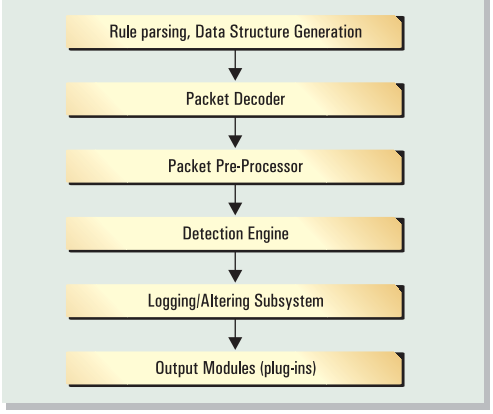


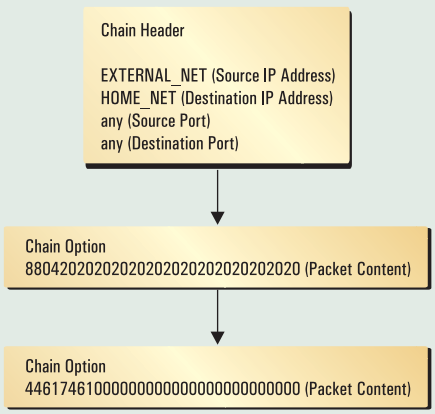
Figure 4

a) Example of Snort rules that differ only in packet content

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING Pinger
Windows";
content:"|4461746100000000000000000000000|";)

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING Seer
Windows";
content:"|88042020202020202020202020202020|";)
```

b) Snort rule chain structure



chain header and chain options. The common attributes—like source address, destination address, source port and destination port—form the chain header. The packet- specific attributes—like packet content, TCP flags, ICMP codes, payload size, etc—form the chain options. So if you specify two rules that differ only in the packet content, then Snort, while parsing those rules, places the common attributes in the chain header and the packet specific attribute (packet content in this case) in the chain options. Figure 4 illustrates this.

Snort can log the network packets to the disk either in human readable format or in tcpdump format. The human readable format is a plain ASCII text format and is expensive because the network packet must be decoded before logging. The tcpdump format is a binary format and is less expensive; it should be used if performance is critical.

Snort raises an alert when the detection engine matches a packet against a rule. It can be configured to either send the alert to the disk, or as a SYSLOG message, or as a Win pop-up message, or to UNIX sockets. The reporting sub-system handles these logging and alerting functionalities.

Snort allows users and programmers to extend its functionality through a modular plug-in architecture. Pre-processing of the network packets and output handling are a few examples of the modular plug-in architecture . HTTP decode, port scan detection, IP de-fragmentation, TCP stream reassembly, are Snort pre-processors. Pre-processing modules are called before the detection engine is called, and after the packet has been decoded. Similarly, logging alerts as SYSLOG messages with custom facilities and priorities, logging in tcpdump format, logging to database, are a few examples of the Snort output modules. To know more about the pre-processors and output modules, you can refer to the Snort user manual.

Snort can operate in three different modes: packet sniffing mode, packet logging mode and NIDS mode.

Packet sniffing mode: As a packet sniffer, Snort sniffs the network packet, decodes it and displays it to the user (on the stdout) in human readable format (ascii text format).

Packet logging mode: As a packet logger, Snort sniffs the network packet and logs it to the disk either in human readable format (ascii text) or in tcpdump format, which can later be decoded with Snort or other packet decoding tools (like Ethereal).

NIDS mode: The most complex mode that Snort can work in is

```
byte_extract.c      fpcreate.o      pcrm.c          snprintf.h
# ./snort  -d -e -v
Running in packet dump mode
Log directory = /var/log/snort

Initializing Network Interface eth0

        --== Initializing Snort ==--
Initializing Output Plugins!
Decoding Ethernet on interface eth0

        --== Initialization Complete ==--

-*> Snort! <*-
Version 2.1.0 (Build 9)
By Martin Roesch (roesch@sourcefire.com, www.snort.org)
02/08-17:54:36.138694 0:50:56:C0:0:1 -> 0:50:56:F1:6C:F1 type:0x800 len:0x4A
192.168.64.1 -> 192.168.64.2 ICMP TTL:128 TOS:0x0 ID:40729 IpLen:20 DgmLen:60
Type:8  Code:0  ID:768  Seq:768  ECHO
61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 70  abcdefghijklmnop
71 72 73 74 75 76 77 61 62 63 64 65 66 67 68 69  qrstuvwabcdefghi

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

02/08-17:54:36.138964 0:50:56:F1:6C:F1 -> 0:50:56:C0:0:1 type:0x800 len:0x4A
```

Figure 5: Snort in packet sniffer mode

```
#
#
# ./snort -b -l ./log
Running in packet logging mode
Log directory = ./log

Initializing Network Interface eth0

        --== Initializing Snort ==--
Initializing Output Plugins!
Decoding Ethernet on interface eth0

        --== Initialization Complete ==--

-*> Snort! <*-
Version 2.1.0 (Build 9)
By Martin Roesch (roesch@sourcefire.com, www.snort.org)
#
```

Figure 6: Snort in packet logging mode

```
# ../snort -r ./snort.log.1076243331 | more
Running in packet dump mode
Log directory = /var/log/snort
TCPDUMP file reading mode.
Reading network traffic from "./snort.log.1076243331" file.
snaplen = 1514

        --== Initializing Snort ==--
Initializing Output Plugins!

        --== Initialization Complete ==--

-*> Snort! <*-
Version 2.1.0 (Build 9)
By Martin Roesch (roesch@sourcefire.com, www.snort.org)
Snort exiting
No run mode specified, defaulting to verbose mode
02/08-17:58:56.853285 192.168.64.1 -> 192.168.64.2
ICMP TTL:128 TOS:0x0 ID:49003 IpLen:20 DgmLen:60
Type:8  Code:0  ID:768  Seq:1536  ECHO
61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 70  abcdefghijklmnop
71 72 73 74 75 76 77 61 62 63 64 65 66 67 68 69  qrstuvwabcdefghi

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Figure 7: Snort decodes packet from libpcap format file

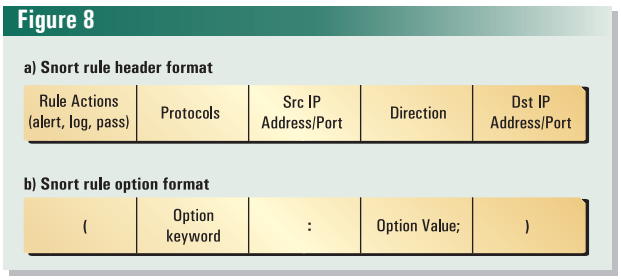
the Network Intrusion Detection Mode (NIDS). As a NIDS, Snort sniffs the network packet, decodes it and analyses the network packet against the set of configured Snort rules. It can do protocol analysis, pattern matching and other detection mechanisms to detect a variety of network attacks.

SNORT RULES

Snort rules form the heart of the Snort system. A rule instructs the detection engine on how to process the network packet.

A Snort rule consists of two sections: rule header and rule options (Figure 8). The rule header contains the rule action, protocol to match, source IP address and port, direction, destination IP address and port. The rule option instructs Snort on what and where to look in a network packet to trigger an action.

Figure 8



Alert, log and pass are the three major rule actions provided by Snort. The alert rule action instructs Snort to generate an alert and then log the packet. The log rule action instructs Snort to just log the packet. The pass rule action instructs Snort to drop the packet. Snort supports TCP, UDP, ICMP and IP in the protocol field of the rule header. There are more than a dozen rule option keywords available in the latest version of Snort. To know more about the rule header and options, you can refer to the Snort user manual.

Some critical actions like blocking traffic, sending TCP resets or ICMP error codes, IP session logging, etc, can be done using the Resp, React, and Tag rule options.

Snort identifies all its rules uniquely by the SID (signature ID) rule option. Numbers that are less than 100 are reserved for future use, and numbers greater than 1000,000 can be used for custom rules.

You can get more information on Snort pre-defined rules from www.snort.org/snort-db.

Snort is a light-weight, easy-to-use intrusion detection system. You can experience its power only by putting it into action, which the second part of this article will cover next time. LFY

The author is a lead engineer working in HCL Technologies (Cisco Systems Offshore Development Centre) in Chennai. He has more than five years of experience in software development including two years in Linux. He holds a B.E. degree in computer science & engineering from S.R.M Engineering College. He can be contacted at rajark_hcl@yahoo.co.in